Analyzing Algorithms: Work and Span

Let T_P be the running time if there are **P** processors available

Two key measures of run-time:

- Work: How long it would take 1 processor = T_1
  - Just "sequentialize" the recursive forking
  - Cumulative work that all processors must complete
- Span: How long it would take infinity processors = T_∞
  - The hypothetical ideal for parallelization
  - This is the longest "dependence chain" in the computation
  - Example: $O($`log` $n)$ for summing an array
    - Notice in this example having > $n/2$ processors is no additional help
  - Also called "critical path length" or "computational depth"

Connecting to performance

Recall: T_P = running time if there are **P** processors available

Work = T_1 = sum of run-time of all nodes in the DAG

- That lonely processor does everything
- Any topological sort is a legal execution
- $O(n)$ for simple maps and reductions

Span = T_∞ = sum of run-time of all nodes on the most-expensive path in the DAG

- Note: costs are on the nodes not the edges
- Our infinite army can do everything that is ready to be done, but still has to wait for earlier results
- $O($`log` $n)$ for simple maps and reductions

Definitions

A couple more terms:

- Speed-up on **P** processors: T_1 / T_P

- If speed-up is **P** as we vary **P**, we call it perfect linear speed-up
  - Perfect linear speed-up means doubling **P** halves running time
  - Usually our goal; hard to get in practice

- Parallelism is the maximum possible speed-up: T_1 / T_∞
  - At some point, adding processors won't help
  - What that point is depends on the span

Parallel algorithms is about decreasing span without increasing work too much

Amdahl's Law (mostly bad news)

Let the ***work*** (time to run on 1 processor) be 1 unit time

Let **S** be the ***portion*** of the execution that can't be parallelized

Then: $T_1 = S + (1 - S) = 1$

Suppose we get perfect linear speedup *on the parallel portion*

Then: $T_P = S + \frac{1-S}{P}$

So the theoretical overall speedup with P processors is (Amdahl's Law):

$$\frac{T_1}{T_P} = \frac{1}{S + (1-S)/P}$$

And the parallelism (infinite processors) is:

$$\frac{T_1}{T_\infty} = \frac{1}{S}$$